EXHIBIT B

Analyzer Component Architecture

Analyzer Component Architecture

1 Introduction

The Analyzer is a software component that sets up and prepares data for analysis using OLAP as well as data mining techniques. It can be thought of as a software "IC" that can be plugged into an existing program by a developer. The architecture of the Analyzer component is shown in Figure 1. As can be seen, the Analyzer does not have a user interface of its own. Thus, it can be plugged into the user interface of an existing program, or a customized user interface can be designed exclusively for it. Either way, it provides a standard set of interfaces so that its functionality can be made use of.

2 Analyzer Modules

The Analyzer component has 8 main modules. These are described in brief below:

2.1 Input Processor Module

The Input Processor Module is responsible for processing user inputs (which could come from a Graphical User Interface, or a Command Line Interface, useful for batch processing), and passing them on to the Data Handler Module. These inputs are of the following types:

2.1.1 Raw Data Inputs

These inputs relate to the data to be processed -- the kind of data (ASCII Text, HTML, Spreadsheet, local database, remote database, real-time data), the name of the data table, etc.

2.1.2 OLAP Inputs

The user can specify the columns of the data to be included for OLAP analysis.

2.1.3 Attribute Focusing Inputs

The user can specify various parameters for data mining using Attribute Focusing -- the Focus Attributes, the Numeric Attribute, the Decision Variable and the cutoff

2.1.4 Clique Inputs

The user can specify whether or not clique analysis is to be performed on the data.

2.1.5 Grammar Inputs

The user can specify the grammar associated with various columns in the data.

2.1.6 Control Inputs

The user can provide various control inputs, such as "Start Data Import", "Stop Data Import", "Start Analysis", "Stop Analysis", etc.

2.1.7 Queries

The user can query the status of various operations, such as "Is Data Import Over", or "Is Data Analysis Over", or "Get Status", or "Get Last Error", or "Get Data Mining Sample Result". (These are provided so that the User Interface designer can provide appropriate feedback to the user)

2.2 *Data Handler Module*

The Data Handler Module acts as the "CPU" of the Analyzer. It controls the imports and analysis of several sets of data. It performs the following tasks:

- Sets up internal data structures according to the inputs provided by the Input Processor Module for each dataset

- Co-ordinates Data Import of a dataset with the Data Access Module, and stores the raw data imported by the Data Access Module

- Co-ordinates analysis of a dataset with the Blazer Module

2.3 *Data Access Module*

This module is responsible for importing different kinds of data into the Analyzer, and passing the data into a message queue for analysis by the Blazer Module. The import functionality of the Blazer functions within a separate thread of control. This is to enable the user to cancel the import operation, if required.

2.4 *Blazer Module*

The Blazer Module is responsible for data analysis. It follows a proprietary data mining algorithm. The analysis functionality also runs within a separate thread of control, so that the user can cancel the operation, if required.

2.5 *Output Processor Module*

The Output Processor Module is responsible for storing the analysis results in a suitable format on the hard disk. It also stores the grammar associated with the raw data. It is controlled by the Blazer Module, and thus operates within the same thread as the analysis function of the Blazer.

2.6 *Example Generator Module*

The Example Generator Module generates sample data mining results, based on the inputs provided to it (the attributes involved, the decision variable, etc.). Thus, the user can get a "quick look" at the kind of results produced based on the inputs she has provided.

2.7 *Error Handler Module*

The Error Handler Module handles all errors encountered by the other modules. It also provides a textual description of the last error encountered. It is responsible for graceful exit, in case of a fatal error.

2.8 *Status Handler Module*

The Status Handler Module is responsible for providing status reports to the user. The other modules constantly feed this module with the latest status of their operations.
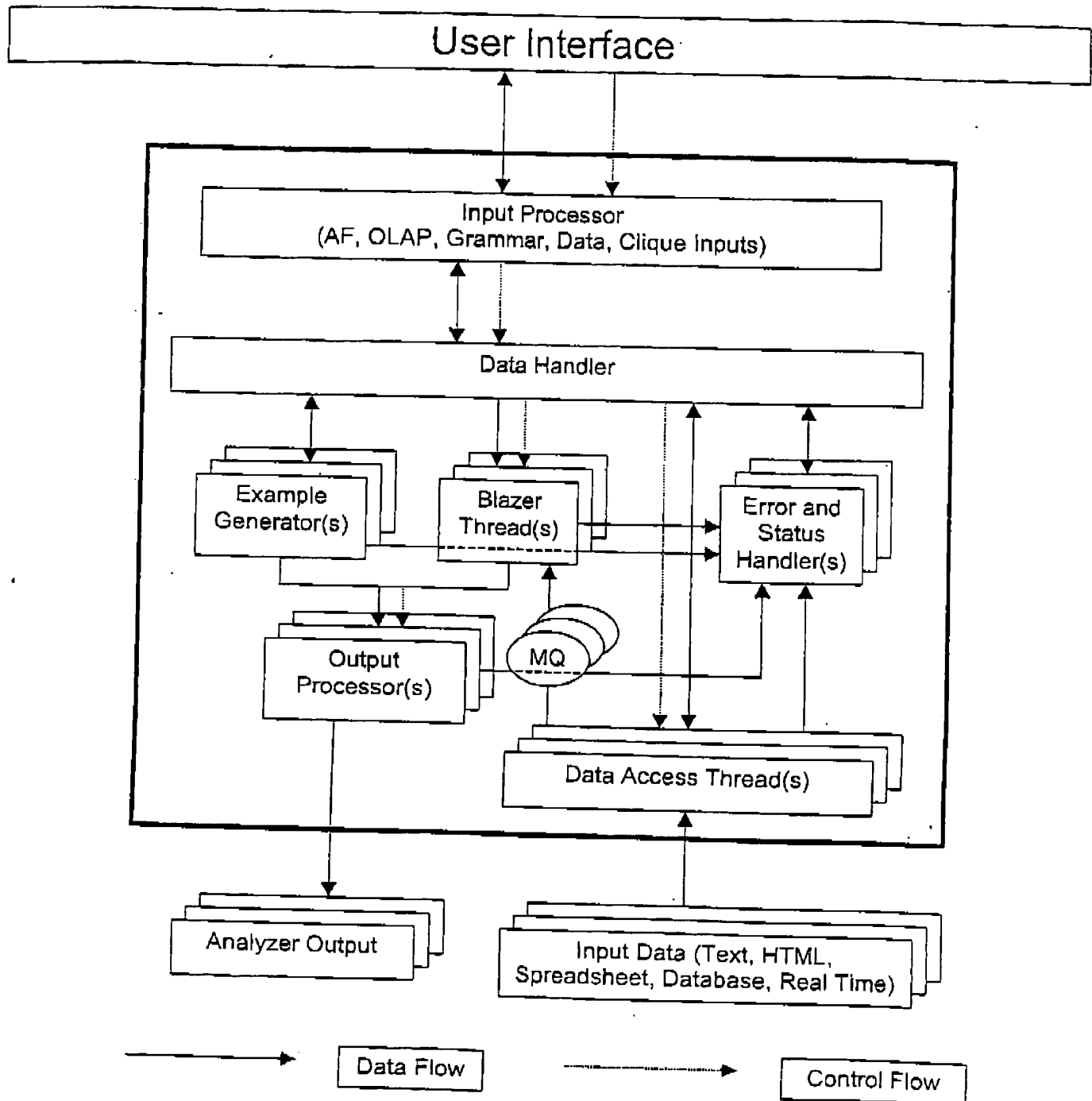
Figure 1: Analyzer Component Architecture

Figure 1: Analyzer Component Architecture